

IN THE CLAIMS

This listing of claims will replace all prior versions, and listings, of claims in the application:

Listing of Claims:

1. (Original) A method for managing an object in memory, comprising:

assigning the object to an assigned frame wherein the object can be released when the assigned frame is released;

detecting an attempt to place a reference to the object in an older frame, the older frame being older than the assigned frame; and

reassigning the object to a reassignment frame that is at least as old as the older frame.
2. (Original) A method for managing an object in memory as recited in Claim 1, wherein the reassignment frame is the older frame.
3. (Original) A method for managing an object in memory as recited in Claim 1, wherein assigning the object to the assigned frame comprises associating a frame identifier with the object.
4. (Original) A method for managing an object in memory as recited in Claim 1, wherein assigning the object to the assigned frame comprises associating a frame identifier with a reference of the object.
5. (Original) A method for managing an object in memory as recited in Claim 1, wherein assigning the object to the assigned frame comprises associating a frame identifier with the object and detecting an attempt to place a reference to the object in an older frame is performed using the frame identifier.
6. (Original) A method for managing an object in memory as recited in Claim 1, wherein detecting an attempt to place a reference to the object in an older frame comprises

comparing a first frame identifier associated with the object with a second frame identifier associated with the older frame.

7. (Original) A method for managing an object in memory as recited in Claim 1, wherein detecting an attempt to place a reference to the object in an older frame comprises comparing a first address associated with the object with a second address associated with the older frame.

8. (Original) A method for managing an object in memory as recited in Claim 1, wherein detecting an attempt to place a reference to the object in an older frame comprises determining whether the object is in stack memory or heap memory.

9. (Original) A method for managing an object in memory as recited in Claim 1, wherein detecting an attempt to place a reference to the object in an older frame comprises determining whether the object is in stack memory or heap memory by examining a distinguishing bit or a distinguishing set of bits.

10. (Original) A method for managing an object in memory as recited in Claim 1, wherein detecting an attempt to place a reference to the object in an older frame comprises determining whether the object is in stack memory or heap memory; and heap memory is uniquely identified by a heap frame identifier.

11. (Original) A method for managing an object in memory as recited in Claim 1, wherein reassigning the object to a reassignment frame that is at least as old as the older frame comprises recursively detecting whether the object references any younger object.

12. (Original) A method for managing an object in memory as recited in Claim 1, wherein reassigning the object to a reassignment frame that is at least as old as the older frame comprises recursively detecting whether the object references any younger object and reassigning any referenced younger object to the reassignment frame.

13. (Original) A method for managing an object in memory as recited in Claim 1, wherein reassigning the object to a reassignment frame comprises resetting a first frame identifier associated with the object to be the same as a second frame identifier associated with the reassignment frame.
14. (Original) A method for managing an object in memory as recited in Claim 1, wherein reassigning the object to a reassignment frame comprises moving the object into the reassignment frame.
15. (Original) A method for managing an object in memory as recited in Claim 1, wherein reassigning the object to a reassignment frame comprises moving the object into the reassignment frame and storing overflow in an overflow area associated with the reassignment frame.
16. (Original) A method for managing an object in memory as recited in Claim 1, wherein reassigning the object to a reassignment frame comprises expanding the reassignment frame and moving the object into the reassignment frame.
17. (Original) A method for managing an object in memory as recited in Claim 1, wherein reassigning the object to a reassignment frame comprises moving the object into the reassignment frame and updating a reference to the object.
18. (Original) A method for managing an object in memory as recited in Claim 1, wherein reassigning the object to a reassignment frame comprises moving the object into the reassignment frame and updating all references to the object.
19. (Original) A method for managing an object in memory as recited in Claim 1, wherein reassigning the object to a reassignment frame is assisted by a display.
20. (Original) A method for managing an object in memory as recited in Claim 1, wherein reassigning the object includes tracing reassigned space.

21. (Original) A method for managing an object in memory as recited in Claim 1, further comprises modifying an allocation site of the object.

22. (Original) A method for managing an object in memory as recited in Claim 1, further comprises modifying a frame creation site.

23. (Original) A method for managing an object in memory as recited in Claim 1, further comprises learning reassignment information.

24. (Original) A method for managing an object in memory as recited in Claim 1, further comprising performing thread-local garbage collection.

25. (Original) A method for managing an object in memory as recited in Claim 1, further comprising storing call path information associated with an allocation site of the object.

26. (Original) A method for managing an object in memory as recited in Claim 1, wherein detecting an attempt to place a reference to the object in an older frame is performed with hardware assist.

27. (Currently Amended) A tangible computer program product for managing an object in memory, the tangible computer program product being embodied in a computer readable medium and comprising computer instructions for:

assigning the object to an assigned frame wherein the object can be released when the assigned frame is released;

detecting an attempt to place a reference to the object in an older frame, the older frame being older than the assigned frame;

reassigning the object to a reassignment frame that is at least as old as the older frame.

28. (Original) A system for managing an object, comprising:

a processor configured to:

assign the object to an assigned frame wherein the object can be released when the assigned frame is released;

detect an attempt to place a reference to the object in an older frame, the older frame being older than the assigned frame;

reassign the object to a reassignment frame that is at least as old as the older frame; and a memory coupled to the processor, configured to provide the processor with instructions.

29. (Cancelled)

30. (Cancelled)

31. (Cancelled)

32. (New) A tangible computer program product for managing an object in memory as recited in Claim 27, wherein the reassignment frame is the older frame.

33. (New) A tangible computer program product as recited in Claim 27, wherein assigning the object to the assigned frame comprises associating a frame identifier with the object.

34. (New) A tangible computer program product as recited in Claim 27, wherein assigning the object to the assigned frame comprises associating a frame identifier with a reference of the object.

35. (New) A tangible computer program product as recited in Claim 27, wherein assigning the object to the assigned frame comprises associating a frame identifier with the object and detecting an attempt to place a reference to the object in an older frame is performed using the frame identifier.

36. (New) A tangible computer program product as recited in Claim 27, wherein detecting an attempt to place a reference to the object in an older frame comprises comparing a first frame identifier associated with the object with a second frame identifier associated with the older frame.

37. (New) A tangible computer program product as recited in Claim 27, wherein detecting an attempt to place a reference to the object in an older frame comprises comparing a first address associated with the object with a second address associated with the older frame.

38. (New) A tangible computer program product as recited in Claim 27, wherein detecting an attempt to place a reference to the object in an older frame comprises determining whether the object is in stack memory or heap memory.

39. (New) A tangible computer program product as recited in Claim 27, wherein detecting an attempt to place a reference to the object in an older frame comprises determining whether the object is in stack memory or heap memory by examining a distinguishing bit or a distinguishing set of bits.

40. (New) A tangible computer program product as recited in Claim 27, wherein detecting an attempt to place a reference to the object in an older frame comprises determining whether the object is in stack memory or heap memory; and heap memory is uniquely identified by a heap frame identifier.

41. (New) A tangible computer program product as recited in Claim 27, wherein reassigning the object to a reassignment frame that is at least as old as the older frame comprises recursively detecting whether the object references any younger object.

42. (New) A tangible computer program product as recited in Claim 27, wherein reassigning the object to a reassignment frame that is at least as old as the older frame comprises recursively detecting whether the object references any younger object and reassigning any referenced younger object to the reassignment frame.

43. (New) A tangible computer program product as recited in Claim 27, wherein reassigning the object to a reassignment frame comprises resetting a first frame identifier associated with the object to be the same as a second frame identifier associated with the reassignment frame.

44. (New) A tangible computer program product as recited in Claim 27, wherein reassigning the object to a reassignment frame comprises moving the object into the reassignment frame.

45. (New) A tangible computer program product as recited in Claim 27, wherein reassigning the object to a reassignment frame comprises moving the object into the reassignment frame and storing overflow in an overflow area associated with the reassignment frame.

46. (New) A tangible computer program product as recited in Claim 27, wherein reassigning the object to a reassignment frame comprises expanding the reassignment frame and moving the object into the reassignment frame.

47. (New) A tangible computer program product as recited in Claim 27, wherein reassigning the object to a reassignment frame comprises moving the object into the reassignment frame and updating a reference to the object.

48. (New) A tangible computer program product as recited in Claim 27, wherein reassigning the object to a reassignment frame comprises moving the object into the reassignment frame and updating all references to the object.

49. (New) A tangible computer program product as recited in Claim 27, wherein reassigning the object to a reassignment frame is assisted by a display.

50. (New) A tangible computer program product as recited in Claim 27, wherein reassigning the object includes tracing reassigned space.

51. (New) A tangible computer program product as recited in Claim 27, the tangible computer program product further comprising instructions for modifying an allocation site of the object.

52. (New) A tangible computer program product as recited in Claim 27, the tangible computer program product further comprising instructions for modifying a frame creation site.

53. (New) A tangible computer program product as recited in Claim 27, the tangible computer program product further comprising instructions for learning reassignment information.

54. (New) A tangible computer program product as recited in Claim 27, the tangible computer program product further comprising instructions for performing thread-local garbage collection.
55. (New) A tangible computer program product as recited in Claim 27, the tangible computer program product further comprising instructions for storing call path information associated with an allocation site of the object.
56. (New) A tangible computer program product as recited in Claim 27, wherein detecting an attempt to place a reference to the object in an older frame is performed with hardware assist.
57. (New) A system as recited in Claim 28, wherein the reassignment frame is the older frame.
58. (New) A system as recited in Claim 28, wherein to assign the object to the assigned frame comprises to associate a frame identifier with the object.
59. (New) A system as recited in Claim 28, wherein to assign the object to the assigned frame comprises to associate a frame identifier with a reference of the object.
60. (New) A system as recited in Claim 28, wherein to assign the object to the assigned frame comprises to associate a frame identifier with the object and to detect an attempt to place a reference to the object in an older frame is performed using the frame identifier.
61. (New) A system as recited in Claim 28, wherein to detect an attempt to place a reference to the object in an older frame comprises to compare a first frame identifier associated with the object with a second frame identifier associated with the older frame.
62. (New) A system as recited in Claim 28, wherein to detect an attempt to place a reference to the object in an older frame comprises to compare a first address associated with the object with a second address associated with the older frame.

63. (New) A system as recited in Claim 28, wherein to detect an attempt to place a reference to the object in an older frame comprises to determine whether the object is in stack memory or heap memory.

64. (New) A system as recited in Claim 28, wherein to detect an attempt to place a reference to the object in an older frame comprises to determine whether the object is in stack memory or heap memory by examining a distinguishing bit or a distinguishing set of bits.

65. (New) A system as recited in Claim 28, wherein to detect an attempt to place a reference to the object in an older frame comprises to determine whether the object is in stack memory or heap memory; and heap memory is uniquely identified by a heap frame identifier.

66. (New) A system as recited in Claim 28, wherein to reassign the object to a reassignment frame that is at least as old as the older frame comprises to detect recursively whether the object references any younger object.

67. (New) A system as recited in Claim 28, wherein to reassign the object to a reassignment frame that is at least as old as the older frame comprises to detect recursively whether the object references any younger object and reassigning any referenced younger object to the reassignment frame.

68. (New) A system as recited in Claim 28, wherein to reassign the object to a reassignment frame comprises to reset a first frame identifier associated with the object to be the same as a second frame identifier associated with the reassignment frame.

69. (New) A system as recited in Claim 28, wherein to reassign the object to a reassignment frame comprises to move the object into the reassignment frame.

70. (New) A system as recited in Claim 28, wherein to reassign the object to a reassignment frame comprises to move the object into the reassignment frame and storing overflow in an overflow area associated with the reassignment frame.

71. (New) A system as recited in Claim 28, wherein to reassign the object to a reassignment frame comprises to expand the reassignment frame and to move the object into the reassignment frame.

72. (New) A system as recited in Claim 28, wherein to reassign the object to a reassignment frame comprises to move the object into the reassignment frame and to update a reference to the object.

73. (New) A system as recited in Claim 28, wherein to reassign the object to a reassignment frame comprises to move the object into the reassignment frame and to update all references to the object.

74. (New) A system as recited in Claim 28, wherein to reassign the object to a reassignment frame is assisted by a display.

75. (New) A system as recited in Claim 28, wherein to reassign the object includes to trace reassigned space.

76. (New) A system as recited in Claim 28, wherein the processor is further configured to modify an allocation site of the object.

77. (New) A system as recited in Claim 28, wherein the processor is further configured to modify a frame creation site.

78. (New) A system as recited in Claim 28, wherein the processor is further configured to learn reassignment information.

79. (New) A system as recited in Claim 28, wherein the processor is further configured to perform thread-local garbage collection.

80. (New) A system as recited in Claim 28, wherein the processor is further configured to store call path information associated with an allocation site of the object.

81. (New) A system as recited in Claim 28, wherein to detect an attempt to place a reference to the object in an older frame is performed with hardware assist.

INTERVIEW SUMMARY UNDER 37 CFR §1.133 AND MPEP §713.04

A telephonic interview in the above-referenced case was conducted on June 20, 2006 between the Examiner and the Applicants' representative Lee Van Pelt. The proposed amendments set forth herein were discussed with the intent to place the claims in better condition for allowance or appeal.

The Applicants wish to thank the Examiner for the interview.